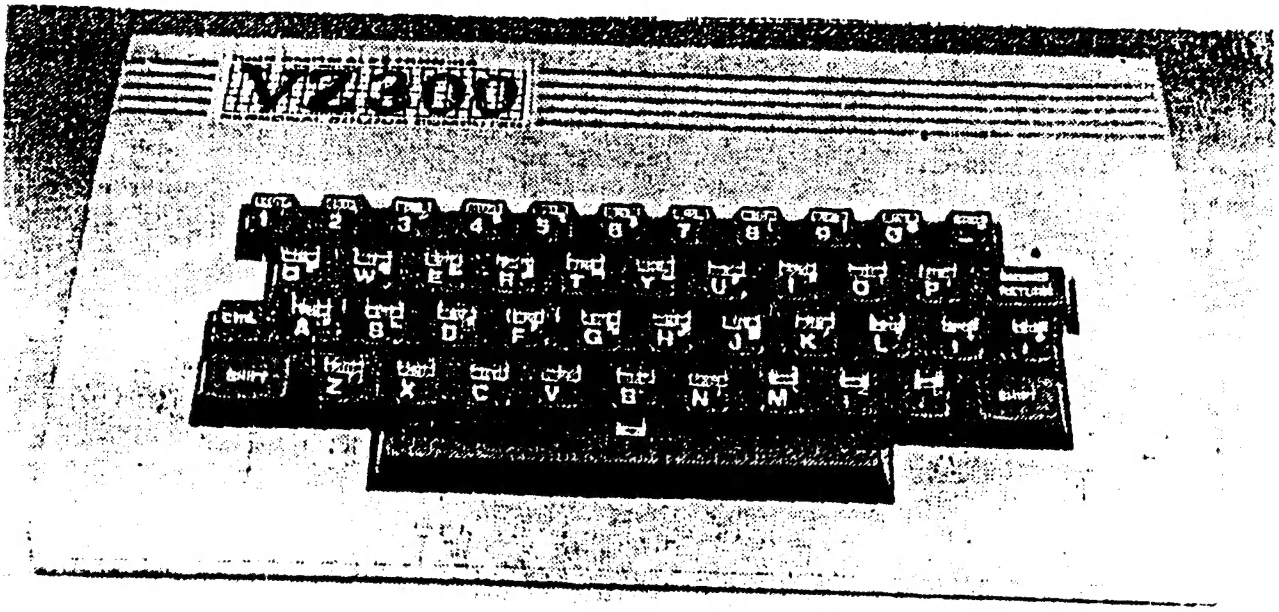
ISSUE NUMBER 6.

HUNTER VALLEY VZ USERS' GROUP MAGAZINE



NOVEMBER 1986.

PRESIDENT......Gavin Williamson

SECRETARIES....Ross woods

....Peter Ellis

TREASURER......Leigh Rogers

GAVIN.....(049) 621678

ROSS......(049) 712643

PETER.....(049) 695697

The Editor,
Hunter Valley VZ Magazine,
Gavin Williamson,
204 Brunker rd,
Broadmeadow. 2292

SECRETARIES. . .

Peter Ellis,
3 Gow St..
Hamilton North 2303

Ross Woods,
83 Lambton Pde.,
Swansea Heads. 2281

EDITORIAL- - -

An exciting month in the development of VZs. I guess the ball really started rolling with Bob Kitch. Bob was the initial force behind all the importing of hardware and software from West Germany. A project that took many long days and seemingly longer nights. I'm certain, that that in the next couple of weeks, the rewards will outweigh all the effort. Interesting to hear, was the positive reaction from D.S.E. I've always said that competion was healthy, and it seems that some of D.S.E people may believe this too. Wouldn't it be nice to see a lowering of price on 'Old Faithful'.

Contact was made with the VZ USER Club in Sydney. I have seen most (some are out of print) of the magazines that Mark Harwood, VZ USERS' Editor, has compiled and I feel that these magazine are well worth subscribing to. Subscription fee is $15 per year. Marks postal address is P.O box 154, DURAL 2158.

Good to see more enquiries to membership coming our way. The most recent being from Orange, NSW, and just prior to this, we now have a couple of members in W.A. Welcome aboard all!

It was great to have the opportunity to entertain Bob Kitch, who was down our way for a couple of days last week. Bobs first article appears in this month's edition. This is an introduction to programming with a difference. Read on, as this feature actually delves into the methodology and architecture of programming rather than being a direct instructional, already obtainable from any bookshop. Incidently, Bob has issued an invitation to one and all involving your programming problems. Don't hesitate to write to him, as I'm sure you will find Bob most helpful. Also included with this Issue is a copy of a list of articles, on the VZ, that Bob has indexed over the last couple of years. Of course our regular features appear within. Thanks to Robert Quinn, Part 4 of Understanding your VZ is with us. Matthew Taylor put in a lot of work this month. Thanks Matthew for the Games and Utility Reviews. For our Technical Report this month, thanks to Dave Mitchell, Dave Boyce and to our Technical Advisor, Joe Leon for redrawing the schematic. Unfortunately, when our magazine gets to this size, some things have to be left out. Sorry folks, but this month it was the printer/plotter and beginners sections. Compensation for the beginners is the article by Bob Kitch and the program on the following page by Paul and Joe Leon. An interesting set of routines, part of which involves PEEK/POKE Editing (see this month Understanding Your VZ, not Token editing, but the theory remains the same).

Things to look out for are: the second article by Bob Kitch in Australia Electronics Monthly on the VZ video display expansion, also in the Oct. Issue is the (not quite as up to date as the one included in this Magazine) list of articles by Bob, on the VZ. Thanks to AEM for showing some more support for the VZ. Also coming from Bob in the next couple of weeks is an eight way expansion board. This unit will plug into your peripheral port on your VZ and enable you to run up to eight (or more if you wish) accesories off your VZ. These could include dot matrix printer, printer/plotter, joysticks, speech synthesiser, compumuse and almost any amount of external control units, for doorbells, burglar alarms, etc... On the topic of looking out for goodies, Joe Leon should (fingers crossed) have his battery backed RAM up and running by tonight. I recently acquired a 64K RAM printer buffer from Don McKenzie. Excellent piece of equipment.

Lastly, you will see a change in leadership. This is because Peter felt that he had achieved what he had planned to do, that was to set up the Club. I feel Pete should be congratulated for his success in tackling this mammoth task. Both Pete and Ross will share the secretarial duties. Also our new meeting procedures take effect . of tonight, in our new hall the New Lambton community centre.

MENU MAKER

This is a simple MENU MAKER for disk drive owners. Type in lines 10 and 20 without alteration and make sure you put the space before the word RUN in line 20.

Commands are as follows:-

: - Pressing inverse key will toggle between normal/inverse screen.

$ - This will give a DIRectory and STATUS to the screen.

# - You can enter filenames in DATA statements after pressing this key. Make sure that line 400 is retained as is. After entering filename use a comma followed by a T for text or a B for binary files. Line 230 will POKE a B in front of RUN in line 20 or a space in case of T, text files while line 240 will POKE the filename in line 20.

^ - Use this key with care, as it will ERAse, SAVE and RUN Menu.

& - If your disk hasn't a Menu then & will SAVE and RUN Menu.

Use POKE 30777,1 with caution as it will hang up the disk drive, but it does speed up printing to the screen. It is wise to POKE 30777,35 before accessing disk drive.

If you own a VZ300 change the POKE 30744,96 in lines 50 and 150 to POKE 30744,1. This has to do with inverse screen toggling.

```
10 GOTO40
20  RUN"MENU     "
30 '"DON'T CHANGE LINES 10 TO 20
40 DIMP$(28),F$(28)    29    29
50 COLOR,0:POKE30744,96:POKE30862,80:POKE30863,52:POKE30777,1
60 N=N+1:READP$(N),F$(N):IFF$(N)<>"*"THEN~~GOTO~~60ELSEP$(N)=""
70 CLS:PRINT"          JOE'S MENU MAKER    :  RW    ";
80 FOR I=1 TO 14
90 PRINT@(32*I),"█"CHR$(191+I)" "P$(I);
100 PRINTTAB(15)"█"CHR$(205+I)" "P$(I+14)TAB(30)"██";
110 NEXT
120 PRINT" $ DIR    # LIST    ^ REPL    & SAVE ";:POKE29183,96
130 GOSUB310
140 X$=INKEY$:X$=INKEY$:IFX$=""THEN140ELSEX=USR(X):POKE30777,1
150 IFX$=":",POKE30744,ABS(PEEK(30744)=0)*96:SOUND20,1
160 IFX$="^"THENGOTO260
170 IFX$="&"THENGOTO270
180 IFX$="$"THENGOSUB280
190 IFX$="#"THEN:CLS:LIST320-
200 IFX$<"@"ORX$>CHR$(N+63)THEN140
210 P=ASC(X$)-63:CLS:PRINT@480,"LOADING "F$(P)": "P$(P);
220 GOSUB310
230 IFF$(P)="B"THENPOKE31477,66ELSEPOKE31477,32
240 FORB=1TO8:POKE31479+B,ASC(MID$(P$(P)+"        ",B,1)):NEXT
250 POKE30777,35:GOTO20
260 POKE30777,35:ERA"MENU":SAVE"MENU":RUN"MENU"
270 POKE30777,35:SAVE"MENU":RUN"MENU"
280 POKE30777,35:CLS:DIR:STATUS:PRINT@18," PRESS RETURN ";
290 R$=INKEY$:R$=INKEY$
300 IFR$<>CHR$(13)THEN290ELSESOUND30,1:RUN
310 SOUND20,1;25,1:RETURN
320 '"     ENTER FILENAMES HERE     "
330 '"     1 TO 8     CHARACTERS     "
340 DATA MENU,T
400 DATA,*
```

UTILITY REVIEW. - -

BY MATTHEW TAYLOR

Hackers Delight is a powerful utility program for the VZ which was sold by Celestron software. (Celestron no longer exists, see Ed Note Issue #1, pg.7). The program is only available on disk for obvious reasons but this is by no means a major barrier to usage.

Upon running the boot program the disk drive loads a colourful and well designed title page. It features Hackers Delight written in red in the sky over the sun and clouds. Below this is a boat anchored in the distance near an island and in front of this is another small island with a large palm tree on it. Foaming blue water surrounds all this.

When any key is pressed the disk drive activates and the central menu is loaded. You are given three options on this menu.

These are 1. Program Copier
2. Dissasembler
3. Z-80 Monitor.

The program copier is basically the same as the old one except for a one added feature. Just as a refresher the options are:-

<L> Load a program
<T> Save a text program
<B> Save a binary program
<P> Program information
<D> Tape to disk transfer
<Q> Return to central menu

Commands L,T,B and Q are self explanitory but the commands P and D need a little explanation.

The <P> command gives information about the program just loaded. This information includes the start and end addresses of the program loaded in decimal and hexadecimal. The amount of memory taken up by the program is given in bytes and kilobytes also. If any key is pressed (other than <M>) you will be asked if you wish to change the start address pointer. If a reply of yes is given then you must input the new address in hexadecimal. It then asks the same thing about the end address before returning to the menu.

The <D> command allows for transfer of binary programs, that have been loaded by tape, to disk. The program stops and ask you to type the following line and press return. BSAVE"program name",ssss,eeee. Substitute the program name for what you want to call it and ssss and eeee for the given start and end addresses. Disk drive owners should be familiar with this process and the use of this DOS command.

The second option of the central menu is the dissasembler. Once loaded the program immediately asks for the start address of disassembly. This is followed by the end address and output to printer (y or n). The program then proceeds to dissasemble memory from the start address onwards into standard nmemonics.

If output to printer is chosen the disassembly will go to the printer as well as to the screen. The command <S> will pause disassembly at any time and <C> will continue it again. The <A> command will abort disassembly and you will be asked if you wish to return to the central menu or go to disassembly.

When disassembly reaches the input end address, the computer will print the total number of bytes and instructions on the screen (If printer is being used it will be printed there as well).

cont.....>

UTILITY REVIEW. . .

The third and final option on the central menu is the Z-80 monitor. The sub-menu for this has options (a-h). These are <A> View memory, <B> Alter memory, <C> Relocate memory, <D> Execute memory, <E> Hex - Decimal, <F> Decimal - Hex, <G> View and search and <H> Return to menu.

I won't go into detail with each but here are some points to remember. All entering is done in hexadecimal. Data is entered viewed in lots of 8 bytes. When altering memory if you wish a byte to stay the same hit return rather than typing it again. Remember to save your program before execution otherwise you may lose it. Make sure you know what you are doing before attempting block memory moves.

Basically the order of tasks in fiddling with a program are to load it with program copier, look at it with dissasembler if you need to, alter it with the Z-monitor and then save it again with program copier. This is a very appropriately named program and is worth getting hold of.

IMPORT REPORT. . .

About mid-september Bob Kitch (QLD, but not naturalised) told me of a contact address in West Germany, a wholesaler who dealt in computer products. Now some six weeks down the track, I'm getting some very excited phone calls from Club Members wondering what stage proceeding are at. To fill those in who may have missed the full story, further contact was established with Ce-Tec Trading of Hamburg, West Germany. Prices were worked out for an extensive range of software and hardware. After many days of negotiations, between banks, the Customs Dept, the Taxation office an Import Agent and Ce-Tec, an order was finally placed for $AUS3832. The order was as follows:-

12	Lightpens
1	Printer/Plotter
2	Printer Interfaces
4	16K RAM Modules
1	64K RAM Module
13	Disk Controllers
14	Disk Drives

Also a comprehensive range of software was ordered.

The last communication I made to Ce-Tec was by 'phone on Thurs the 30th. of Oct. That was to inform them that the bank in Australia had completed all it had to do. I was then assured that all goods were packed (order sent by Fax 2 days prior) and ready to be freighted.

So now all we do is wait. Thanks to Bob Kitch for doing the original groundwork and to all those who supported this scheme through its initiation. This will probably be offered again in March next year.

TECHNICAL REPORT. - -

BY DAVE MITCHELL.   SCHEMATIC BY JOE LEON

The following is another hardware modification to your VZ. Two points to remember, these alterations should not be tried by those who are not confident with a soldering iron and secondly, any physical modification to your computer will void all warranties.

Use this when a program crashes or to 'BREAK' most Machine Code programs, for resetting the Computer without turning the System OFF and ON.

When the power is turned on, the Soft Start is in the RESET state. When the switch is pressed, No#1 (gate) goes HIGH and NO#2 (gate) goes LOW (refer to the diagram). The 1 Meg resistor discharges the 100n cap to switching point, then Soft Start is reset. When the switch is released, the 100n cap recharges ready for another go.

Also when the switch is pressed, NO#1 puts 5 volts onto the NMI (non maskable interupt) for a few seconds. The Computer then jumps to 0066hex, this is where the subroutine for the interupt lives.

Resetting the Disk Drive with the Soft Start. Ever tried to RUN, BRUN, LOAD, BLOAD, SAVE or BSAVE a program to or from Disk without a Disk in the Drive or with the Door open ? This is where the Soft Start can be used also.

Leave the Door open on the drive after you fit the Soft Start and type RUN"H", the Computer will be locked up for ever unless you turn it OFF and ON, but I don't like the noises that come from the Drive.

Instead, I press the Soft Start, this returns control to me, but the Drive is still going hell for Mary (it's still running, sorry about that Mary) so I have to turn OFF the Drive.

There are TWO (2) other ways -:

1 - turn Off & ON the Power to the Drive, which is the easiest way - or -

2 - for the hell of it !! use Software.

The following two programs do work but it is just as easy to turn the power to the Drive OFF & ON. In the New VZ 300 TECH. Manual there is a section devoted to the Disk Drive.

At 4008 Hex is a subroutine to turn ON the Drive and at 400B Hex is a subroutine to turn OFF the Drive. Also in the the 311 Bytes for housekeeping is a Byte that the controller looks at to see if the Drive is turned OFF or ON and if Drive 1 or 2 has been selected.

So, from this I have written two programs, one in Basic and one in Assembler Code.

The BASIC Program.

```
1 POKE 30862,11: Poke 30863,64: X=USR(0)
2 POKE PEEK(30897)+256* PEEK(30898)+11-65536,0
3 POKE 30862,8: POKE 30863,64:X=USR(0)
4 POKE PEEK(30897)+256* PEEK(30898)+11-65536,16
5 STOP
```

Line 1 turns the Drive OFF

Line 2 works out where the housekeeping lives and sets the Byte to 0 (zero).

Line 3 turns the Drive on.

Line 4 resets the housekeeping to Drive 1

If we didn't use line 2, the Drive would start running again when it executed line 3.
Instead of lines 2 & 4 with the TOP of MEMORY at 65535 (FFFF Hex), you can change them to POKE -300,0 and POKE -300,16 (that's if your mem goes that high).
If you have a VZ-300 with a Standard 16 K Ram Pack then these POKEs would be -

```
    line 2 POKE-2348,0  and
    line 4 POKE-2348,16.
```

Assembler Code Program.
Does the same as the Basic program.

```
CALL 400B H         ;turn off drive
LD   HL,(79B1H)     ;load HL with top of memory
LD   DE,11          ;load DE with 11 decimal
ADD  HL,DE          ;add them
LD   A,0            ;load A with zero
LD    (HL),A        ;load A into the address pointed to
                        by the HL register
PUSH HL             ;save HL on stack
CALL 4008 H         ;turn disk on
POP HL              ;get HL from stack
LD A,16             ;load A with 16 decimal
LD (HL),A           ;load A into address pointed to
                        by the HL registers
JP 1A19 H           ;jump to Basic
```

Both programs or either program could be saved on TAPE and used when necessary.
But again, it is easier to turn the Power to the Drive OFF and On.

SOFT START INTERRUPT from DAVE MITCHELL

PARTS LIST

1 × 4093

1 × 10n Greencap

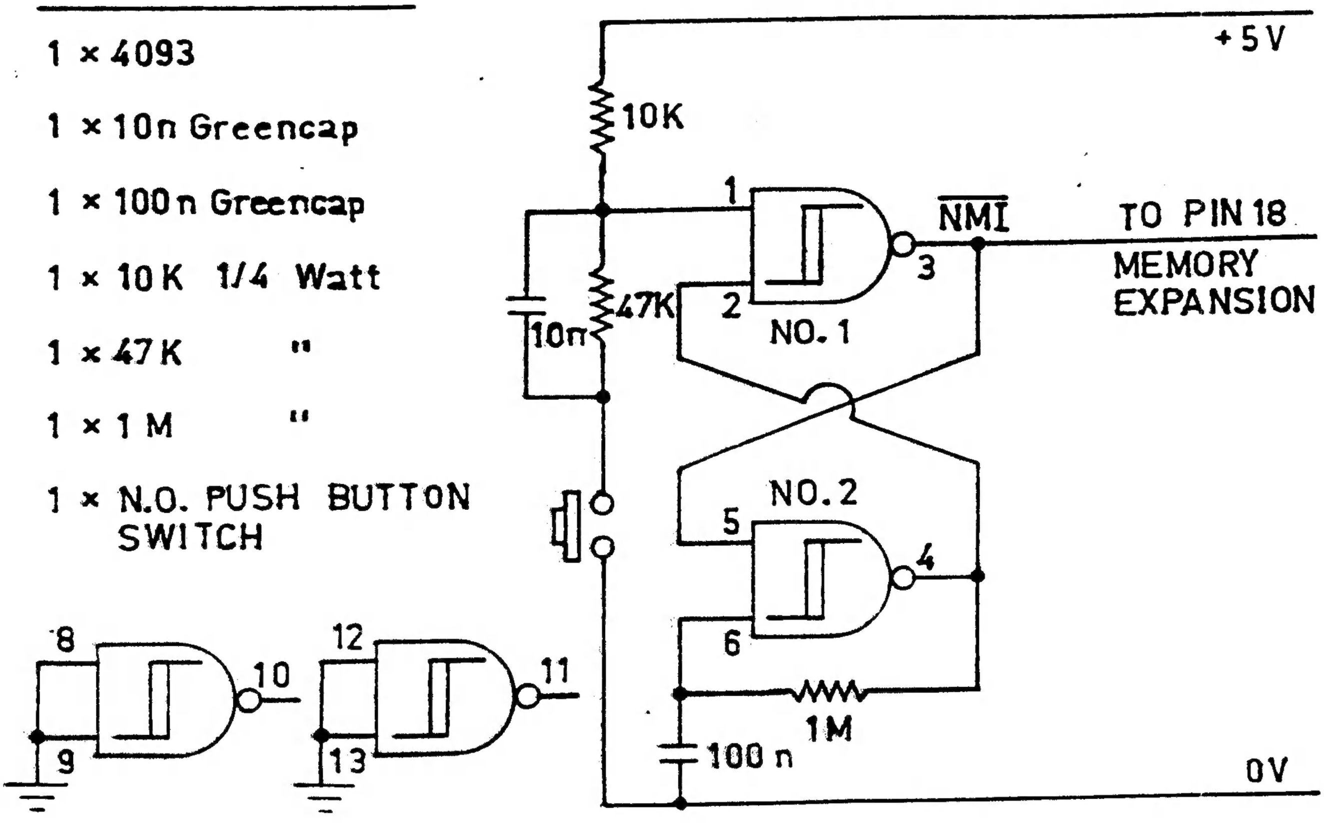
1 × 100n Greencap

1 × 10K 1/4 Watt

1 × 47K "

1 × 1 M "

1 × N.O. PUSH BUTTON SWITCH



INTRODUCTION TO PROGRAMMING

PART 1 - BY BOB KITCH

I have been asked to contribute a series on BASIC programming for the HV-Users. So here goes.

Firstly, the series will be unconventional. Most introductions to BASIC proceed blow-by-blow through the various BASIC commands. I will not - many texts exist which can explain these better than I can.

Secondly, the series will initially be non-specific to any particular computer language. General programming concepts and guidelines will be offered. The principles will be equally applicable to BASIC, Assembler, Pascal or whatever.

Thirdly, advanced programming concepts and hints will be offered as they are needed. This is the best time to introduce these since their mystique is removed.

Fourthly, early emphasis will be on PLANNING, ORGANISING and MAINTAINING a program, rather than encouraging feverish coding at the keyboard (which is usually commenced too early by beginners).

It is quite possible to recognise a breed of compulsive programmers, born from the home micro boom. This breed, is emerging from the brave new world of tomorrow's technology whose reason for existence is simply to program. People become totally fascinated by the unlimited abstract world that the inside of a computer offers. We can create a Universe or any World inside a machine. In the abstract world of programming, a well thought out programming method serves as a MAP, and the techniques of software engineering are the WEAPONS. These then are the main threads of this series.

Let's commence this month with a few definitions and concepts to ponder over until the next installment.

THE COMPUTER is a machine, and is only capable of doing simple work. It has been termed by some as "a remarkably efficient counting machine with a large memory - but no brains!" It has no intelligence and cannot think.

A COMPUTER SYSTEM consists of four elements :-

1. the Central Processor Unit (in the VZ it is the Z-80A microprocessor chip) with "primary memory" (ROM and up to 34K RAM).

2. Input devices - keyboard, cassette, disk and so on.

3. Output devices - screen, printer, cassette, disk, in-built speaker, voice and sound synthesisers etc.

4. "Secondary memory" - not essential but may be cassette or disk when used to update or relieve primary memory.

MAN-MACHINE INTERFACE. The interaction between man-machine inputs and outputs is a continuous and circular feedback process. e.g. man output (keypress) is machine input ..or.. machine output (screen prompt) is man input- ..and so on. This interaction forms the basis of using computers.

The four fold subdivision of a computer system is little different to our own mental capabilities. The CPU and primary memory is broadly equivalent to our mind. The I/O devices are similar to our senses (touch, taste, sight, sense of heat, speaking, hearing). The secondary memory is directly comparable to our use of external aids to assist our memory, such as note books, filing cabinets of information, telephone directories - all of which have slow access and are difficult to recall compared with things already resident in our mind.

COMPUTER PROCESSES or CAPABILITIES are surprisingly few in number. There are only FOUR and unless an exercise or problem can be broken down into these elementary processes, then coding of the program should not commence. A greater understanding of the problem is required before proceeding.

It is important to clearly distinguish two things whilst programming. The first, is to devise a LOGICAL solution to the programming exercise, which is quite independent of the particular language to be used. The second, is the actual CODING of the exercise being undertaken. The latter stage is easy, provided that the former is well understood. The computer program will only function correctly if the logic of the program is correct, and there are no aids or diagnostics available from the machine to assist in achieving correctness in this demanding aspect of program design. Some diagnostics are however available to assist in the coding portion of the task - such as the SYNTAX checking.

As one becomes more familiar with programming languages it is soon apparent that many of the powerful command structures are simply macro instructions formed from these few "primatives".

The four processes are :-

1. Input data and store it in primary memory - the data may be either "raw" data input (e.g. from keyboard) or read-in from the secondary store. (e.g.tape).

2. Output data already stored in primary memory - either as "output" (e.g. to screen) or written-out to secondary memory (e.g.tape).

3. Perform simple arithmetic procedures (addition or subtraction) upon data in primary memory only.

4. Perform logical comparisons (disjunction, conjunction and negation) between two items of data in primary memory.

(Remember - I/O, arithmetic, comparisons only)

To continue the analogy with ourselves, I doubt whether we can do anything more than these operations except that we use experience. The computers' analogue of this is the PROGRAM as it possesses zero intelligence.

THE PROGRAMMING TASK is to utilize the high speed and large memory capacity of a computer system to do something useful - such as carry out calculations (number crunching), play games, monitor house security etc.

The spectrum of tasks involved in programming is very broad, so little wonder that beginners have trouble grasping the essentials, or that many programs are "badly" written. The task involves taking an idea or concept and translating that into a symbolic (program statement) form of representation. An intermediate stage in this translation often involves modelling the phenomenon being programmed. This psychologically involves moving from concrete concepts to various levels of abstraction - again a very difficult thing for, particularly young, minds to master.

The transition from an idea to a program can seldom be achieved in one leap - more often a number of intermediate steps are required. Liken it to writing an essay where drafts and notes are used before the final prose is produced. Fortunately a number of useful tools have been developed to assist in producing a good program.

In my view, one of the greatest pitfalls of the home computer boom is that these intermediate steps are not understood by Users so that, at least, bad programs and, at worst, disillusioned programmers result. Many of these people may find their way into the computer industry of the future. There is ALWAYS more personal satisfaction in achieving a "good" job even if it is only a games program for the kids. It is also more fun, (the essence of home micros) as there is less hassle in getting a program to run, and more time for more programs.

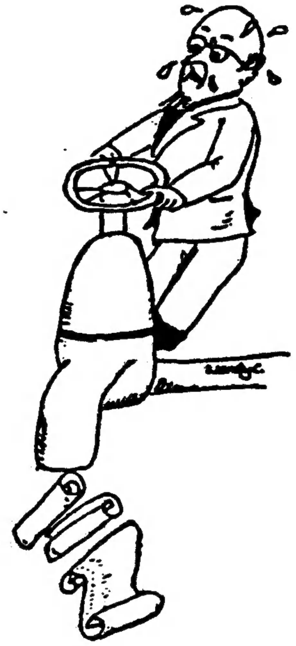
In the microcomputer environment where there are always hardware limitations, it means that it is very difficult to completely seperate hardware and software aspects of the programming task. The programmer may have to get "close to the hardware" - usually due to hardware/memory limitations or restricted I/O capabilities. Don't shy away from hardware by saying "but I am only interested in writing programs" as the two are somewhat inseperable.

Next month we will look at the various stages in the programming task, or how to approach a programming exercise. (see, no mention of BASIC code in this article!)

Finally, I would like to offer to Users that your programming queries will be answered if you write to me - with a SAE. please. In this manner you should get what you want and I will obtain a feel for the type of problems Users in the Hunter Valley are experiencing.

WRITE TO:-

BOB KITCH,
7 EURELLA ST.,
KENMORE. QLD. 4069.



Running programs

GAMES REVIEW- - -
BY MATTHEW TAYLOR

Starblaster is one of the older Dick Smith games. Price $14.95. The object of the game is to shoot and destroy all of the enemy's fleet in your area of the galaxy.

The first screen is the standard joysticks (y or n) screen. After selection a title screen comes up and singly, the letters in the word starblaster, expand onto the screen. A rather interesting and creative effect. This is then followed by a fighter which comes closer and closer toward you, beeping as it comes. Once it gets right up to you, (really big), the screen goes into a brilliant beeping, multi-coloured explosion sequence.

Next is the first page of instructions which explain the object of the game and level selection details. The second page tells you about the enemy fleet and the types of craft that you'll encounter during the game. The third screen displays the control keys. The game has five levels of play (1-5), five being the hardest. Once a level is selected the game imediately begins.

At the start of the game you are given 999 units of energy. Energy is used three ways :- when firing, if you are hit by a fighter and at a constant rate to stay in flight. There is a yellow frame around the outside of the screen and in the middle of this is a small cross. This cross is your aiming device and when a enemy craft passes in front of this, and you fire, the enemy will be destroyed. When you fire, shots come from each corner of the screen and meet in the middle. If you hit an enemy craft a colourful and musical (but quick) explosion sequence will take place. The game rather resembles T.V. POW on NBN 3 on saturday mornings.

In Starblaster you are given the impression that you can control, to an extent, the enemy craft's movement and ultimately steer it into the weapon sights. In fact, this is done by manouvering your ship about and so changing your view of the area. To begin with this can be a little confusing, as the enemy ships will move the opposite to what direction you press, but remember you are moving your view of your craft, not the actual enemy ships. Amongst all this drama are little wiggling stars to comfort you.

Whilst navigating the galaxies, be careful because as with the title page, the enemy gradually gets closer and closer to you. If you do not destroy them in time, they will score a direct hit on you and depending on your level, you will lose a certain amount of energy to absorb the hit.

There are four types of enemy craft. These are freighters (FHT), tankers (TNK), troop carriers (TRP) and fighters (FHT). The higher your level the more fighters and the greater the rate of energy consumption for firing, hit absorbtion and flight maintainance. Here is a table of the numbers of each type of enemy craft for each level. The right hand columns of figures show energy consumption for firing and being hit at each of the five levels.

Level	FRT	TNK	TRP	FHT	FIRING	HIT ABSORB
1	5	7	4	4	5	50
2	4	7	4	5	8	60
3	4	6	4	6	10	70
4	4	5	4	7	15	80
5	4	4	3	9	20	100

This game, written by Dubois and Mc Namara is in smooth scrolling high resolution graphics and can get quite addictive after a while.

If this doesn't happen, the simplest remedy is to look at Adams' score on level one in the top scores section of this magazine (below) and you soon will be. Last but not least are the control keys and until next month, happy shooting.

Q ...	SCAN UP
A ...	SCAN DOWN
M ...	SCAN LEFT
, ...	SCAN RIGHT
Z or SPACE ...	FIRE LAZERS
(1-5) ...	SELECT LEVEL

GRAPHICS	****
SOUND	***
ANIMATION ...	***
OVERALL	***

Loading times:- TAPE 2 mins
- DISK 13 secs

HIGH SCORE TABLE

ASTERIODS	35020	MATTHEW TAYLOR
CIRCUS	1080	ADAM MAGEE
DAWN PATROL	60200	MATTHEW TAYLOR
DIG OUT	29200	MATTHEW TAYLOR
GALAXON	18780	CHRISTIAN WARNER
GHOST HUNTERS	13640	ADAM MAGEE
HAMBURGER SAM	47400	MATTHEW TAYLOR
HOPPY	10740	MATTHEW TAYLOR
LADDER CHALLENGE	25400	MATTHEW TAYLOR
PANIK	10890	MATTHEW TAYLOR
PLANET PATROL	987	ADAM MAGEE
ROAD WARRIOR	28370	MATTHEW TAYLOR
SPACE INVADERS	17290	MATTHEW TAYLOR
STAR BLASTER	812	ADAM MAGEE
SUPER SNAKE	1183	EDITOR

Might have a canidate for gamblers anonymous. Stephen Taylor, Matthews young brother won $4,052,903 on the D.S.E POKER. Informed sources tell me that the only reason Stephen didn't keep on winning was because his prizes were getting so big, he was having trouble working out the standard notation. What a problem!